Generic Hash Functions

- Java's hash tables are generic
  - Can store elements of arbitrary type
- To support elements of arbitrary type
  - Hash function must be able to map objects to integers
  - Hash table lookup must be able to compare objects for equality

hashCode() and equals()

- hashCode()
  - Defined in java.lang.Object, the first ancestor of all Java classes
  - Returns an integer value to put into the hash function
  - By default, returns the memory address of the object
- equals()
  - Also defined in java.lang.Object
  - Used while iterating over objects in the same bucket
  - By default, checks if the memory addresses are equal

API Contracts

- Given two objects o1 and o2
- Rules
  - Symmetry: If o1.equals(o2), then also o2.equals(o1)
  - If o1.equals(o2), then o1.hashCode() == o2.hashCode()
- Have to be obeyed for hash tables to work correctly
  - If you override equals(), you must also override hashCode()

```java
public class StudentRecord {
   private long id;
   private String firstName;
   private String lastName;

   public StudentRecord(long id, String firstName, String lastName) {
      this.id = id;
      this.firstName = firstName;
      this.lastName = lastName;
   }

   public boolean equals(Object o) {
      if (!(o instanceof StudentRecord))
         return false;
      StudentRecord other = (StudentRecord)o;
      return id == other.id;
   }

   public int hashCode() {
      return (int)((id >> 32) ^ id);
   }

   public String toString() {
      return id + ": " + firstName + " " + lastName;
   }
}
```

Example

```
import java.util.HashSet;

public class Test {
    public static void main(String[] args) {

     HashSet<StudentRecord> enrolled = new HashSet<StudentRecord>();

     StudentRecord stu = new StudentRecord(1234, "Jane", "Doe");
     StudentRecord stu2 = new StudentRecord(2000, "John", "Foo");

     enrolled.add(stu);
     enrolled.add(stu2);
     System.out.println(enrolled); // John and Jane

     StudentRecord stu3 = new StudentRecord(2000, "Jack", "Black");
     enrolled.add(stu3);   // Jack has same id as John, not added
     System.out.println(enrolled); // Still John and Jane
    }
}
```